

REMARKS

Claims 29 and 40-50 have been amended. No claims have been added or cancelled. Claims 1-50 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

37 CFR 1.75(c) Objection:

The Examiner objected to claims 39 and 50 under 37 CFR 1.75(c) as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicants respectfully traverse this objection for at least the following reasons.

Claim 39 is dependent from, and further limits the subject matter of, claim 29, in contrast to the examiner's contention. For instance, claim 29 recites tracking *mutable accesses* of a plurality of attributes on a client state of session data. Claim 39 recites tracking *accesses of mutable attributes* of the client state of the session data. Thus, while claim 29 is concerned with tracking mutable accesses (e.g. accesses that modify or change an attribute), claim 39 is concerned with tracking accesses (not necessarily just mutable accesses) of mutable attributes. Thus, claim 39 does provide further limitation on the subject matter recited in claim 29. Similar arguments apply to claim 50 as well.

Double Patenting Rejections:

The Examiner rejected claims 1 and 14 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1 and 7 of pending U.S. Patent Application No. 10/087,652. The Examiner also rejected claims 1, 14, 26, 29 and 40 as being unpatentable over claims 1, 7, 15 and 21 of pending U.S. Patent Application No. 10/087,224 and claims 1, 7, 15 and 21 of pending U.S. Patent Application No. 10/087,225. Claims 1, 14, 29 and 40 as being unpatentable over claims 1, 8 and 9 of pending U.S. Patent Application No. 10/087,197. Claim 26 as being unpatentable over claims 1, 10, 17, 26 of pending U.S. Patent Application No.

10/087,234. The instant application and the 10/087,652, 10/087,224, 10/087,225, 10/087,197, 10/087,234 applications are all pending patent applications, not issued patents. If and/or when any of these rejections become non-provisional, Applicants will consider filing a terminal disclaimer or present reasons traversing the rejection.

Section 101 Rejection:

The Examiner rejected claims 29-50 under 35 U.S.C. § 101 as directed to non-statutory subject matter. Applicants have amended claims 29 and 40-50 to overcome this rejection. As such respectfully request removal of the § 101 rejections of claims 29-50.

Section 103(a) Rejections:

The Examiner rejected claims 26 and 27 under 35 U.S.C. § 103(a) as being unpatentable over Courts (U.S. Patent 6,360,249) in view of Zaiken (U.S. Patent 5,907,848). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 26, contrary to the Examiner's assertion, Courts in view of Zaiken fails to teach or suggest means for an application server to track mutable accesses of the attributes in a client state of session data. Instead, Courts teaches that session data is downloaded from global session server 220 and provided to render engine 212 (*see, e.g.* Courts, column 8, lines 15-24 and lines 59-64) and that changed data from a session cache is written back to global session data. Court is silent with regard to tracking mutable accesses of the attributes in a client state of session data. Zaiken also fails to mention anything regarding tracking mutable accesses of attributes in the client state of the session data and thus fails to overcome the deficiencies of Courts regarding tracking mutable accesses of attributes in the client state of the session data. Thus, Courts and Zaiken, both singly and in combination fail to teach an application server configured to track mutable accesses of the attributes in the client state of the session data.

In response to the above arguments, the Examiner, in the Response to Arguments, again cites column 5, lines 20-50 and column 8, lines 1-25 of Courts. However, neither of the Examiner's cited passages teaches tracking mutable accesses of attributes in a client state of the session data. Instead, the first cited passage (column 5, lines 20-50) describes Courts' use of dynamic load balancing "to evenly distribute requests across any number of web system servers." The Examiner argues that Courts' monitoring of client requests for load balancing purposes and Courts' session cache teach tracking mutable accesses of attributes in a client state of session data. The Examiner has incorrectly interpreted the teachings of Courts. Courts does mention anything about tracking accesses of mutable attributes as part of dynamic load balancing. Monitoring received requests, such as web page requests, does not involve tracking of mutable accesses of attributes. Load balancing of requests to servers does not include or imply any tracking of mutable accesses of attributes in a client state of session data. Courts teaches that URL requests are monitored and distributed to servers using any of various routing and distribution algorithms, such as round robin (column 6, lines 17-26).

The second cited passage (column 8, lines 1-25) describes Courts' caching of session data. Specifically, Courts teaches that a session cache within a session manager locks a SID associated with a client request and downloads session data for that SID from a global session server. After Courts' render engine has built a responsive web page and provided it to the user, the session cache unlocks the SID and writes the changed session cache data to the global session data. Presumably the Examiner contends that any reference to updating global session from a session cache includes tracking accesses of mutable attributes. However, merely writing cached session data back to global session data does not in any way include or imply tracking accesses of mutable attributes in the client state of the session data. Traditionally, accesses of mutable attributes in session data are not tracked and global session data is overwritten with the data from a session cache. Courts does not describe tracking accesses to individual mutable attributes in the session cache.

Moreover, the Examiner's interpretation of Courts is clearly incorrect because the

Examiner is relying on to disparate functions in Courts that have nothing to do with one another. Specifically, the Examiner refers to Courts' monitoring of client requests in support of load balancing across multiple servers to teaching tracking, while referring to Courts' session caches to teach mutable session data (See Office Action dated June 27, 2005, page 8, lines 10-12). However, Courts' monitoring of request for load balancing has nothing to do with the session caches nor about writing the session cache data back to the global session data described at the Examiner's cited passage (column 8, lines 1-25). Thus, the Examiner contends that two separate and independent portions of Courts' system somehow teach the specific limitation of tracking mutable accesses of a plurality of attributes of a client state of session data. Such an interpretation is clearly incorrect.

Additionally, Courts in view of Zaiken does not teach or suggest determining a set of mutably accessed attributes of the client state of the session data of an application server. The Examiner cites column 7, lines 60-67 where Courts describes how his render engines service a user request. Courts teaches that his render engines can get current state information for a session through a session manager and session cache. Courts' render engines then process requests using the obtained state information. Nowhere does the cited passage mention anything about determining a set of mutably accessed attributes of the client state of an application server. Instead, as noted above, the cited passage merely describes that Courts' render engines can obtain current state and session information. Furthermore, Zaiken also fails to teach or suggest determining a set of mutably accessed attributes of a client state of session data and thus fails to overcome the above noted deficiencies of Courts.

The Examiner contends that Courts teaches a subset of the set of mutably accessed attributes in a client state of session data that are modified in respect to a primary state of the session data. However, the Examiner failed to cite any portion of Courts in support of this contention. The Examiner merely asserts (Office action, dated June 27, 2005, page 8, lines 16-17 and page 9, lines 3-5) that Courts' system includes a subset of modified attributes. The Examiner also admits that Courts fails to teach determining changes between the session cache and global session data. If, as the

Examiner admits, Courts does not determine changes between the session cache and the global session data, it is unclear how Courts system can include a subset of the set of mutably accessed attributes that are modified in respect to a primary state of the session data. Without some clear teaching from Courts describing a subset of a set of mutably accessed attributes that are modified in respect to a primary state of session data, the Examiner's contentions amount to nothing more than hindsight speculation.

The Examiner also contends that Zaiken teaches the comparison of current and previous versions of data in a session to determine the changes and cites column 10, lines 1-30 of Zaiken. However, the cited passage does not describe comparing current and previous versions of data to determine changes in session data. Instead, Zaiken describes, at the cited passage, determining whether a template exists having a user-specified template ID (see also, column 9, lines 1-3 and lines 26-33). Specifically, Zaiken teaches that if a template exists with the same template ID, Zaiken's system determines whether a currently retrieved log record represents a further transaction associated with the template or not. Zaiken determines this by retrieving data fields from the database that correspond to data values stored in the log record and comparing the retrieved data values to the key value for the template (having the specified template ID). If any of the data values match the key value, Zaiken's program adds the filename from the log record to the filenames in the template if it is not already represented in the template. Thus, the Examiner's cited passage does not have any relevance to determining a subset of the set of mutably accessed attributes that are modified in respect to a primary state of session data. Instead, Zaiken is describing searching through log records and comparing data values in both the log records and templates to determine whether a log record corresponds to an earlier transaction or whether it represents a new transaction.

Furthermore, the Examiner's proposed combination of Courts and Zaiken would not result in a system including means for an application server to track accesses of mutable attributes in the client state of session data, determine a set of mutably accessed attributes of the client state of the session data of the application server, and determine a subset of the set of mutably accessed attributes that are modified in respect to the primary

state of the session data. Instead, a combination Courts and Zaiken would result only in a system that performs the load balancing and global session data locking as described in Courts, and includes the template creating ability of Zaiken. Since, as noted above, neither Courts nor Zaiken teaches anything regarding tracking accesses of mutable attributes in a client states of session data, determining a set of mutably accessed attributes, and determining a subset of the set of mutably accessed attributes that are modified in respect to the primary state of session, no combination of Courts and Zaiken would include such functionality.

For at least the reasons discussed above, the rejection of claim 26 is not supported by the prior art and removal thereof is respectfully requested.

The Examiner rejected claims 29-36, 39-47 and 50 under 35 U.S.C. § 103(a) as being unpatentable over Courts in view of Challenger (U.S. Patent 6,026,413).

Regarding claim 29, contrary to the Examiner's assertion, Courts in view of Challenger fails to teach or suggest tracking mutable accesses of a plurality of attributes in a client state of session data. Instead, Courts teaches only that session data is downloaded from global session server 220 and provided to render engine 212 (*see, e.g.* Courts, column 8, lines 15-24 and lines 59-64) and that changes to a data cache are copied back to the global session data. Courts is silent with regard to tracking mutable accesses of the attributes in the client state of the session data.

The Examiner cites column 5, lines 20-50 and column 8, lines 1-25 of Courts. However, neither of the Examiner's cited passages teaches tracking mutable accesses of attributes in a client state of the session data. Instead, the first cited passage (column 5, lines 20-50) describes Courts' use of dynamic load balancing "to evenly distribute requests across any number of web system servers." The second cited passage (column 8, lines 1-25) describes how Courts' caching of session data. Specifically, Courts teaches that a session cache within a session manager calls a "lock" on a SID associated with a

client request and downloads session data for that SID from a global session server. After Courts' render engine has built a responsive web page and provided it to the user, the session cache unlocks the SID and writes changes back from the session cache to the global session data.

The Examiner argues that Courts' monitoring of client requests for load balancing purposes and Courts' session cache teach tracking mutable accesses of attributes in a client state of session data. However Courts does not mention anything about tracking accesses of mutable attributes in the client state of the session data as part of monitoring client requests. Instead, Courts teaches a hub that monitors and distributes requests to available servers in order to provide dynamic load balancing. Load balancing of requests to servers does not include or imply tracking of mutable accesses of attributes in a client state of session data. Courts teaches that URL requests are monitored and distributed to servers using any of various routing and distribution algorithms, such as round robin (column 6, lines 17-26). Nowhere does Courts describe tracking mutable accesses of attributes of a client state of session data.

Presumably the Examiner contends that any reference to updating global session from a session cache includes tracking accesses of mutable attributes. However, merely writing cached session data back to global session data does include or imply tracking accesses of mutable attributes in the client state of the session data. Traditionally, accesses of mutable attributes in session data are not tracked and global session data is overwritten with the data from a session cache. Furthermore, writing data from a session cache back to the global session data in Courts only implies that the final version of data in the cache is written back to the global session data. There is nothing in Courts that teaches or suggests the tracking of mutable *accesses* of attributes. Additionally, Challenger also fails to teach tracking mutable accesses of a plurality of attributes of a client state of session data, and thus does not overcome the above noted deficiencies of Courts.

Additionally, as described above regarding claim 26, the Examiner's interpretation of Courts cannot be correct. Specifically, the Examiner refers to Courts' monitoring of client requests in support of load balancing across multiple servers to teaching tracking, while referring to Courts' session caches to teach mutable session data (See Office Action dated June 27, 2005, page 8, lines 10-12). However, Courts' monitoring of request for load balancing has nothing to do with the session caches nor about writing the session cache data back to the global session data described at the Examiner's cited passage (column 8, lines 1-25). Thus, the Examiner contends that two separate and independent portions of Courts' system somehow teach the specific limitation of tracking mutable accesses of a plurality of attributes of a client state of session data. Such an interpretation is clearly incorrect.

The Examiner admits that Courts fails to teach performing an object graph comparison of mutably accessed attributes with a benchmark version of the client state of the session data to determine a subset of modified attributes, wherein the benchmark version of the client state of the session data comprises a previous version of the attributes in the client state of the session data. The Examiner relies upon Challenger and argues that Challenger teaches an object graph comparison of current and previous version of state data and cites column 10, lines 36-43 and column 28, lines 1-7. Challenger teaches the use of object dependence graphs in order to determine how different are two different caches of a single data source. In other words, Challenger is concerned with propagating changes made to a central data source to multiple data caches. (See, e.g. column 3, lines 22-27; column 9, lines 47-61; column 10, lines 14-24). While Challenger does teach using object dependence graphs for comparing different versions of data, Challenger has nothing to do with comparing mutably accessed attributes with a benchmark version of the client state of the session data to determine a subset of modified attributes. Challenger in combination with Courts does not teach or suggest performing an object graph comparison *of mutably accessed attributes with a benchmark version of the client state of the session data to determine a subset of modified attributes*.

For at least the reasons presented above, the rejection of claim 29 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks apply to claim 40.

The Examiner rejected claims 37, 38, 48 and 49 under 35 U.S.C. § 103(a) as being unpatentable over Courts in view of Challenger and in view of Jackson (Publication No. 2003/0051145). Applicants traverse the rejection of claims 37, 38, 48 and 49 for at least the reasons presented above regarding their respective independent claims.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

Allowed Claims:

Claims 1-25 have been allowed by the Examiner.

Claims Objected To But Otherwise Allowable:

Claim 28 was objected to as being dependent up on a rejected base claim, but otherwise allowable if rewritten in independent form.

CONCLUSION

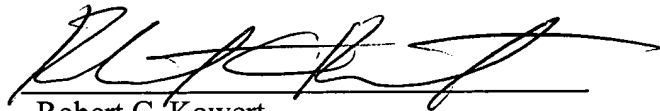
Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Applicants hereby petition for such extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-07800/RCK.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☐ Petition for Extension of Time
- ☐ Notice of Change of Address
- ☐ Information Disclosure Statement

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: September 27, 2005